03-06-00

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:

John J. Burns

Serial No.: N/A Examiner: Unknown

Filing Date: Herewith Group Art Unit: Unknown

For: METHOD FOR EMULATING MULTI-PROCESSOR ENVIRONMENT

Docket No.: 33012/284/101

TRANSMITTAL SHEET

Assistant Commissioner for Patents
Washington, D.C. 20231

Sir:

CERTIFICATE UNDER 37 C.F.R. 1.10: The undersigned hereby certifies that this paper or papers, as described herein, are being deposited in the United States Postal Service, "Express Mail Post Office to Addressee" having an Express Mail mailing label number of : EL 522 531 772 US, in an envelope address to: Assistant Commissioner for Patents, Washington, D.C. 20231 on this 3rd day of March, 2000

By _____
Carolyn L. Erickson

We are transmitting herewith the attached Patent Application including the following:

[XXXX] 16 sheet(s) of specification.

[XXXX] 4 sheet(s) of claim(s).

[XXXX] 1 sheet(s) of Abstract.

[XXXX] 9 sheet(s) of drawings.

[XXXX] Executed Declaration and Power of Attorney.

[ ] A verified statement(s) to establish small entity status under 37 C.F.R. 1.9 and/or 1.27 is enclosed.

[XXXX] An Assignment of the invention to Unisys Corporation is being filed contemporaneous with this patent application.

[ ] A certified copy of a _____ application, serial no. _____, filed _____, 19___, the right of priority of which is claimed under 35 U.S.C. 119.

CLAIMS AS FILED						
	(1)	(2)	SMALL ENTITY		OTHER	
FOR:	# FILED	# EXTRA	Rate	Fee	Rate	Fee
BASIC FEE				$345		$690
TOTAL CLAIMS	20-20 =	0	x9=	$	x18=	$ 0
INDEPENDENT CLAIMS	4 -3 =	1	x39=	$	x78=	$ 78
( ) MULTIPLE DEPENDENT CLAIM PRESENTED			+130=	$	+260=	$ 0
TOTAL			$		$768.00	

*If the difference in Column (1) is less than zero, enter "0" in Column 2.

[XXXX] Other Recordation Form Cover Sheet-Patents Only.

[XXXX] Checks in the amounts of $768.00 and $40.00 are enclosed.

[XXXX] Please charge any deficiencies or credit any overpayment in the enclosed fees to Deposit Account 14-0620.

By: Lawrence M. Nawrocki
Lawrence M. Nawrocki
Reg. No. 29,333

NAWROCKI, ROONEY & SIVERTSON, P.A.
Suite 401, Broadway Place East
3433 Broadway Street N.E.
Minneapolis, Minnesota 55413
Telephone: (612) 331-1464
Facsimile: (612) 331-2239

METHOD FOR EMULATING MULTI-PROCESSOR ENVIRONMENT

CROSS REFERENCE TO CO-PENDING APPLICATIONS

The present application is related to co-pending Patent Application Serial No. 33012/283/101, filed _____, entitled "A Method of Dispatching Target Instructions Using C Code", assigned to the assignee of the present invention and incorporated herein by reference.

BACKGROUND OF THE INVENTION

1. **Field of the Invention** - The present invention generally relates to digital data processing systems, and more particularly relates to emulation of a first device through the use of a second and different type of device.

2. **Description of the Prior Art** - It is well known in the prior art to design and build general purpose instruction processors having a predefined software architecture and instruction set. It is also known to design and build instruction processors having predefined software architectures and instruction sets which are more specialized in nature. Such specialized designs are typically directed toward optimization of a particular task or a closely related group of tasks.

One of the tasks to which a specialized design is particularly pertinent is in the field of real time communications processing. Whereas this application does not require complex arithmetic and mathematical calculations, for example, it does require rapid switching of data and control. It also relies upon rapid transfer of data from one location to another. As a result, a communications instruction processor typically has extensive switching, timing, and data transfer instruction, with only modest attention to complex mathematics, such as floating point arithmetic. In a typical implementation, this means that the communications processor would

have a modest set of general purpose instructions (e.g., add, shift, compare, etc.) along with a number of specialized instructions for queue manipulation, segment handling, buffer allocation, etc.

Perhaps the best example of a communications instruction processor having an especially effective design is the Distributed Communications Processor (DCP) available from Unisys Corporation. The DCP instruction processor is found in many systems in use today. The unique software architecture which provides the efficiencies for this specialized application provides difficulty in upgrading to newer and faster hardware.

It is normal to expect that newer and faster instruction processors are continuously being designed and built. Yet for economic reasons of scale, these designs typically have a more generalized software architecture to promote a larger volume of applications over which to amortize the non-recurring development costs. Thus, to benefit from these newer instruction processors, the existing DCP communications software would need to be reprogrammed to operate within the more generalized software architecture of the new instruction processor.

Such reprogramming has two major disadvantages. First, the cost of the reprogramming process may be substantial and occupy a rather extensive development schedule. Second, programming of the communications software to operate within a more generalized architecture does away with the efficiencies to be realized by the more specialized communications architecture.

The typical answer to this dilemma is emulation. Emulation is the process whereby the host machine (in this case a new instruction processor design having a somewhat generalized software architecture) is programmed to provide the more specialized software architecture of the target machine (in this case the DCP). Using this technique, the existing DCP communications

software can run on a newer instruction processor with an apparently incompatible software architecture. U.S. Patent No. 5,794,011, issued to Paul et al., discusses the technique of emulation.

Thus, emulation most easily resolves the first of the problems by eliminating the expense and long development schedule involved in reprogramming. However, the question of runtime efficiencies is always present when emulation techniques are employed. Unless the specific emulation system is optimized for efficiency, the new and faster host instruction processor may actually executed the desired software more slowly than the older specialized target processor. The easiest way to provide efficient emulation is to provide the emulation logic in microcode. As such, the host machine becomes the target machine via the microprogramming. Unfortunately, this directly changes the host environment to look like the target environment. As a result, the host machine loses its character and can no longer execute code specifically designed for its native environment.

In order to provide an emulation wherein the host machine presents both the target environment and its native environment, it is common for prior art emulation techniques to perform a runtime software comparison within the host machine to locate those host machine instructions required to implement a particular target instruction. For emulation of the DCP, this would require an average of about 150 comparisons per specialized target instruction to be emulated. This would render a hopelessly inefficient emulation.

Of particular difficulty is the emulation of a multi-processor target environment. This difficulty is exacerbated when utilizing a single processor host.

SUMMARY OF THE INVENTION

The present invention overcomes many of the disadvantages found in the prior art by providing a method of and apparatus for software emulation of a multi-processor target on a single processor host machine while continuing to provide the native host environment. Unlike prior art systems which utilize comparisons to select the host instructions required to emulate a particular specialized target instruction, the present invention uses the target instruction as a direct pointer to locate the necessary host instructions.

In accordance with the preferred mode of the present invention, a Pentium microprocessor chip, available from Intel Corporation, is used as the host for emulating the Distributed Communications Processor (DCP), available from Unisys Corporation. The invention utilizes a fast index to the target instruction code used by the processor object. An array of procedures is defined in the application within a header file. Each array is defined as a pointer. The pointer is initially void. The real values are filled in by the constructor of the processor object during the initialization of the processor. Using this array, the processor flow is directed to the single instruction using the op code and an additional four bit field as a direct index to the routine.

When a target instruction is processed by the processor object, it is immediately dispatched using the op code as an index into an array of target instruction pointers. This is an efficient alternative to running a string of comparisons between the op codes and target instructions.

This immediate dispatch of the processor object means that the target instruction is promptly emulated without the time consuming comparison found in the prior art. Thus, the run time execution of the emulator becomes substantially faster and more efficient.

A separate object is utilized for each processor to be emulated. Thus, in emulating a multi-processor environment, a plurality of objects are used wherein each object corresponds to a different target processor. Emulation of a multi-processor target environment by a single processor provides significant flexibility in implementing the emulation system. Obviously, this permits the total multi-processor emulation to be accomplished in a single processor system. Furthermore, in multi-processor host systems, the multi-processor target may be emulated from a single or less than all of the host processors which oftentimes gives a much more efficient division of labor amongst processors.

BRIEF DESCRIPTION OF THE DRAWINGS

Other objects of the present invention and many of the attendant advantages of the present invention will be readily appreciated as the same becomes better understood by reference to the following detailed description when considered in connection with the accompanying drawings, in which like reference numerals designate like parts throughout the figures thereof and wherein:

FIG. 1 is a conceptual diagram of a typical legacy environment employing the DCP:

FIG. 2 is a conceptual diagram of modern network architectures;

FIG. 3 is a conceptual diagram of a modern network integrated with the DCP;

FIG. 4 is a diagram showing the basic DCP architecture;

FIG. 5 is a diagram showing an application implementing the DCP system

FIG. 6 is a an illustration of the operation of the DCP;

FIG. 7 is a detailed diagram showing linking of the host and target instructions in accordance with the present invention;

FIG. 8 is detailed diagram showing operation of a multiprocessor system; and

FIG. 9 is a detailed diagram showing I/O operations.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Fig. 1 is a conceptualized diagram showing a legacy system 10 employing the DCP. This basic legacy system is typical of installed data management systems of various enterprises of differing sizes. Channel connect 12 represents the direct coupling of Host 14 to DCP 18 via cable 16. In the normal situation, Channel connect 12 shows that Host 14 and DCP 18 are physically collocated. In this installation, DCP 18 functions as a data concentrator and interface to single function terminals 22 and 24. These single function terminals are not collocated with the other components of legacy system 10 but are coupled via dedicated link 20, as shown. Use of DCP 18 in this way permits modification and upgrade of single function terminals 22 and 24, along with dedicated link 20, without impact on host 14 and its contained legacy software base.

Fig. 2 is a conceptualized diagram 26 of the various components of modern network architectures. Host A 28 is representative of a typical Unisys Model 2200 host configuration currently available from Unisys Corporation. Fiber network, FDDI Network 30, couples the various components which are not physically collocated with Host A 28. Remote components 38 and 40, coupled via path 34, may be a combination of single function terminals, "intelligent" terminals, other computers, data concentrators, servers, etc.

Unix Host 36 is shown as a component providing a different applications environment. X.25 Network 32 is shown as a component providing a different networking protocol. Novell LAN 44, shows a local area network which couples industry compatible, personal computers 60 and 62. This configuration may be found in the typical small office.

Host B 50 is typically a legacy host available from IBM Corporation. It couples with SNA Network 52 to Token Ring 54. Multiple personal computers, servers, and concentrators may be coupled to and serviced by Token Ring 54. TCP/IP Device 42 shows yet another networking protocol in servicing components 56 and 58 via path 46. Components 56 and 58 may be terminals, computers, servers, etc.

The challenge in integrating these various networks and systems is that which is addressed by the Distributed Communication Processor (DCP) available from Unisys Corporation. This challenge is met through the use of an architecture having a hybrid instruction set with both general purpose and special purpose instructions.

Fig. 3 is a conceptualized diagram 62 showing integration of diverse networks and systems through utilization of the DCP technology. The heart of this integrated system is DCP 600 Series 76, as shown. It couples to the remaining system components via the native communication protocol of each. Thus, each of the system components is functionally interconnected without the need to appreciably modify any of these components to accommodate the differing communication protocols utilized by the other system components.

For example, TCP/IP Device 70 communicates with DCP 76 in accordance with its defined protocol. That permits Ethernet LAN devices 78 and 82, coupled to TCP/IP Device 70 via Ethernet LAN 80, to communicate with DCP 76 and therefore with any device coupled to DCP 76. Similarly, UNIX Host 64 communicates with DCP 76 according to its communication protocol.

In like manner, Host A 66, typically a Model 2200 system available from Unisys Corporation, is functionally coupled to DCP 76. Thus, devices 72 and 74, coupled to Host A 66 via FDDI Network 68, can readily communicate with UNIX Host 64, TCP/IP Device 70, and all devices coupled to DCP 76. In addition, Host B 86, typically and IBM mainframe, along with its SNA Network 88 and Token Ring 90, communicate with DCP 66 in its preferred protocol. And Novell LAN 84, along with LAN devices 94 and 96, communicate with DCP 66 as if it were a Novell LAN device.

Fig. 4 is a block diagram of the basic DCP hardware architecture. Communication Processor 98 is the primary instruction processor of the DCP system. It is this element which is emulated in accordance with the present invention as described in detail below.

Communication Processor 98 is coupled to Local Storage 102 via bi-directional memory bus 100, as shown. Similarly, bi-directional memory bus 104 couples Local Storage 102 to Input/Output Processor 106. Line Modules 118, 120, 122, 124, 126, 128, 130, and 132 provide the individualized electrical and functional interface to the various diverse networks within the system (see also Fig. 3). Mass Storage 116 provides the required large storage capacity. Communication lines 108, 110, 112, and 114 couple directly to various diverse components of the integrated system.

Fig. 5 is a block diagram of the application implementation of the emulated DCP system. In accordance with the preferred mode of practicing the present invention, a COM32 microprocessor, available from Intel Corporation, is utilized as the host processor. The preferred programming language is C++. The applications run under the Windows NT operating system. This configuration permits direct use of such software as Telcon, thus directly providing a number of the diverse system interfaces.

CP Implementation 134 is the application associated with emulation of the DCP architecture. DCP DLL 136 provides the instruction sequences as discussed in detail below. The DCP input/output control is performed by IOP Implementation 142, with PP DLL 144 providing the coding sequences. DCP Line Modules 146, 148, and 150 are thus DCP style interfaces coupling to a number of diverse components employing a variety of network protocol standards.

Winsock or Handler 138 provide the opportunity to interface directly from the Host (i.e., Intel) Native environment under Windows NT control to system components communicating with compatible interfaces. Driver 140 is typically a Windows NT Driver. NIC or TCP/IP 152 couples directly to compatible communication links.

Fig. 6 is a detailed diagram showing the communications processor applications of the preferred emulation technique of the present invention. Initialization 154 is utilized to preset operating conditions. Processor 156 contains Dispatcher 158, Storage Access Methods 160, and Registers 162.

Instructions 170 contains the host instructions required to implement a particular target instruction. Typical instructions include Load 172, Jump 174, Arithmetic 176, Store 178, additional instructions 180, and 300 additional instructions 182. Dispatcher makes the selection as shown and the responses include providing an corresponding instruction to Dispatcher 158 or a requested memory operation via Storage Access Methods 160.

When a memory operation is required, Storage Access Methods 160 requests the operation from Main Storage and Locks 168. Dispatcher 158 preserves its real time processing native environment as shown by Real Time Clock 166 and Events 164. Such indications are communicated to Dispatcher 158 as shown.

Fig. 7 is a detailed diagram showing how the array of procedures 184 points directly to the corresponding instructions 186. The instructions are decoded using the op code and a 4-bit field. These two fields are combined to point directly to the appropriate one of instructions 186.

For example, Instr. (N) 188 is decoded to point to Load 190, and Instr. (N+1) 192 is decoded to point to Jump 194. Similarly, Instr. (N+2) is decoded to point to Arithmetic 198, and Instr. (N+3) 200 is decoded to point to Store 202. Additional instructions 204 are decoded to point to their corresponding instructions 206.

Fig. 8 is a detailed diagram showing emulation of a multiprocessor DCP configuration, which incorporates the preferred mode of the present invention. Even in a multiprocessor host configuration, a single host processor may be utilized to emulate multiple target processors. In this embodiment, a plurality of application processors, including Processor 0 214 through Processor n 218, are defined. Each of the processors operates as previously discussed (see also Figs. 6 and 7). Thus, Processor 0 214 has a Dispatcher 220, Storage Access 22, and Registers 224. Similarly, Processor n 218 has a Dispatcher 230, Storage Access 232, and Registers 234. DCP Storage and Storage Locks 212 operates as previously described for single processor emulation. Instructions 216 are as previously described.

In a typical example, Dispatcher 220 of Processor 0 214 combines an op code and corresponding four bit field producing an index to uniquely specify Add Register 240. Access 236 is made to Add Register 240. In response Request 238 to access the necessary data is made to Storage Access 222 which in turn formats Request 226 to DCP Storage and Storage Locks 212.

In a second example, Dispatcher 230 of Processor n 218 combines an op code and corresponding four bit field which uniquely identifies Compare Constant 246. Access 242 transfers the index to Instructions 216 and Compare Constant 246 is accessed. Request 244 is made of Storage Access 232, which in turn formulates Request 228 for accessing DCP Storage and Storage Locks 212.

Fig. 9 is a detailed diagram showing operation of an emulated input/output instruction. This operates similar to emulation of communications processor instructions. IOP Object 250 contains three major components. Dispatcher 256 and Storage Access Methods 252 operate similar to the corresponding components in the communications processor applications (see also Figs. 6 and 7). Sequencer 258 operates to sequence transmissions which are often handled in specialized hardware in the target machine. These operations include sequencing block transmissions, managing serial transmissions, etc. Sequencer 258 drives applications for individual input/output channels including PP Chain 1 278, PP Chain 2 280, and PP Chain n 282 via path 276.

Main Storage and Locks 248 operates as previously described. Instructions 274 are specialized input/output instructions. These include, Queue 264, Start Data 266, Throttle 268, Jump 270, and Move Data 272.

In the present example, Dispatcher 256 combines the present op code and corresponding four bit field to produce an index to uniquely address Start Data 266. The access is made via path 262. In response, Start Data 266 communicates with Storage Access Methods 252 to make the necessary memory accesses via path 260. Storage Access Methods 252 makes the memory accesses via path 254 coupled to Main Storage and Locks 248.

Having thus described the preferred embodiments of the present invention, those of skill in the art will readily appreciate that the teachings found herein may be applied to yet other embodiments within the scope of the claims hereto attached.

What is claimed is:

CLAIMS

1. In a data processing system having a first processor with a first software architecture, the improvement comprising:

a. a plurality of emulation objects each executable by said first processor wherein each of said emulation objects emulates operation of a different one of a plurality of target processors wherein each of said plurality of target processors has a software architecture different from said first software architecture.

2. An improvement according to claim 1 wherein each of said emulation object is compatible with said first software architecture.

3. An improvement according to claim 2 wherein at least one of said plurality of emulation objections further comprises an array of procedures compatible with said first software architecture and a list of instructions compatible with a second software architecture.

4. An improvement according to claims 3 wherein said list of instructions includes specialized instructions for communications processing.

5. An improvement according to claim 4 wherein each of said array of procedures corresponds to a one of said list of instructions through the use of an operation code and corresponding four bit field.

6. An apparatus comprising:

a. a first instruction processor having a first software architecture; and

b. a plurality of emulation objects responsively coupled to said first instruction processor wherein each of said plurality of emulation objects permits said first instruction processor to emulate a different one of a plurality of target processors and wherein each of said plurality of target processors has a software architecture different from said first software architecture.

7. An apparatus according to claim 6 further comprising a first computer program having a first plurality of instructions which are compatible with said first software architecture.

8. An apparatus according to claim 7 wherein said first plurality of instructions further comprises a specialized communication instruction.

9. An apparatus according to claim 8 wherein a first one of said emulation objects further comprises an array of procedures and a list of instructions.

10. An apparatus according to claim 9 wherein each of said procedures of said array of procedures is directly linked to a different one of said list of instructions.

11. A method of emulating a plurality of target processors by a first processor having a first software architecture incompatible with the software architectures of said plurality of target processors, the method comprising:

a. executing a first emulation object corresponding to a first of said plurality of target

processors; and

b. executing another emulation object corresponding to another of said plurality of target processors.

12. A method according to claim 11 further comprising repeating steps b for each of said plurality of target processors.

13. A method according to claim 12 wherein said first emulation object further comprises a specialized instruction.

14. A method according to claim 13 wherein said specialized instruction further comprises an instruction for communication processing.

15. A method according to claim 14 wherein said first emulation object further comprises an array of procedures.

16. An apparatus comprising:

a. means having a first software architecture for executing computer instructions compatible with said first software architecture; and

b. means responsively coupled to said executing means for containing a plurality of emulation objects wherein each of said plurality of emulation objects corresponds to a different one of a plurality of target processors and each of said plurality of target processors has a software architecture which is incompatible with said first software architecture.

17. An apparatus according to claim 16 wherein a first of said emulation objects further comprises an array of procedures.

18. An apparatus according to claim 17 wherein said first of said emulation objects further comprises a list of instructions wherein each of said array of procedures corresponds to a different one of said list of instructions.

19. An apparatus according to claim 18 wherein at least one of said list of instructions further comprises a communication processing instruction.

20. An apparatus according to claim 19 wherein said list of instructions further comprises a plurality of communication processing instructions.

ABSTRACT OF THE DISCLOSURE

A method of and an apparatus for performing efficient software emulation of a multi-processor target computer by a host computer. The software technique permits multiple processors to be emulated by a single processor The use of software emulation permits the host computer to execute both host programs and target programs. The software emulation is made particularly efficient by utilizing the operation code combined with a separate four bit field to directly address the corresponding host instructions.

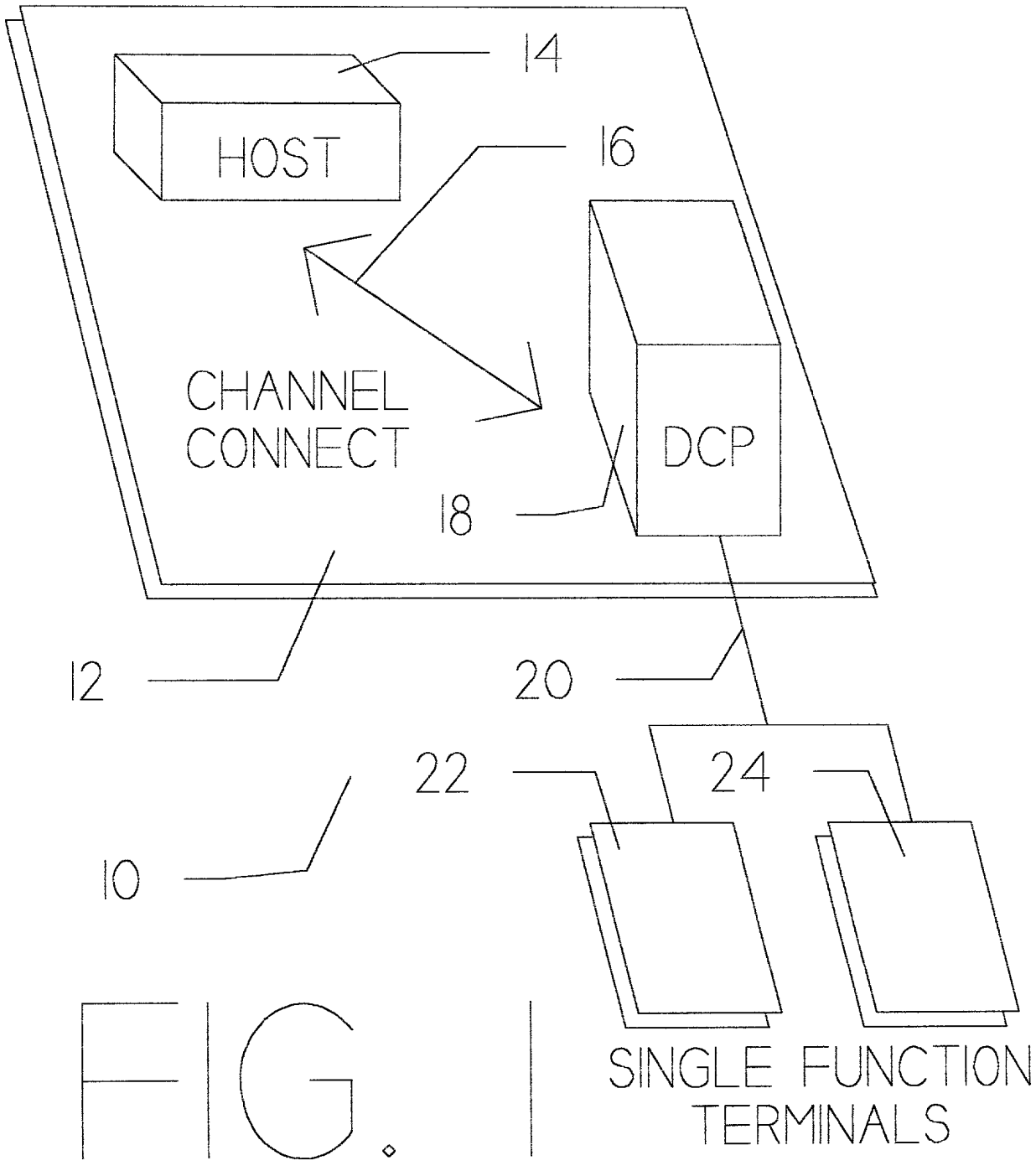
14
HOST
16
CHANNEL
CONNECT
DCP
18
12
20
22
24
10
SINGLE FUNCTION
TERMINALS

FIG. 1

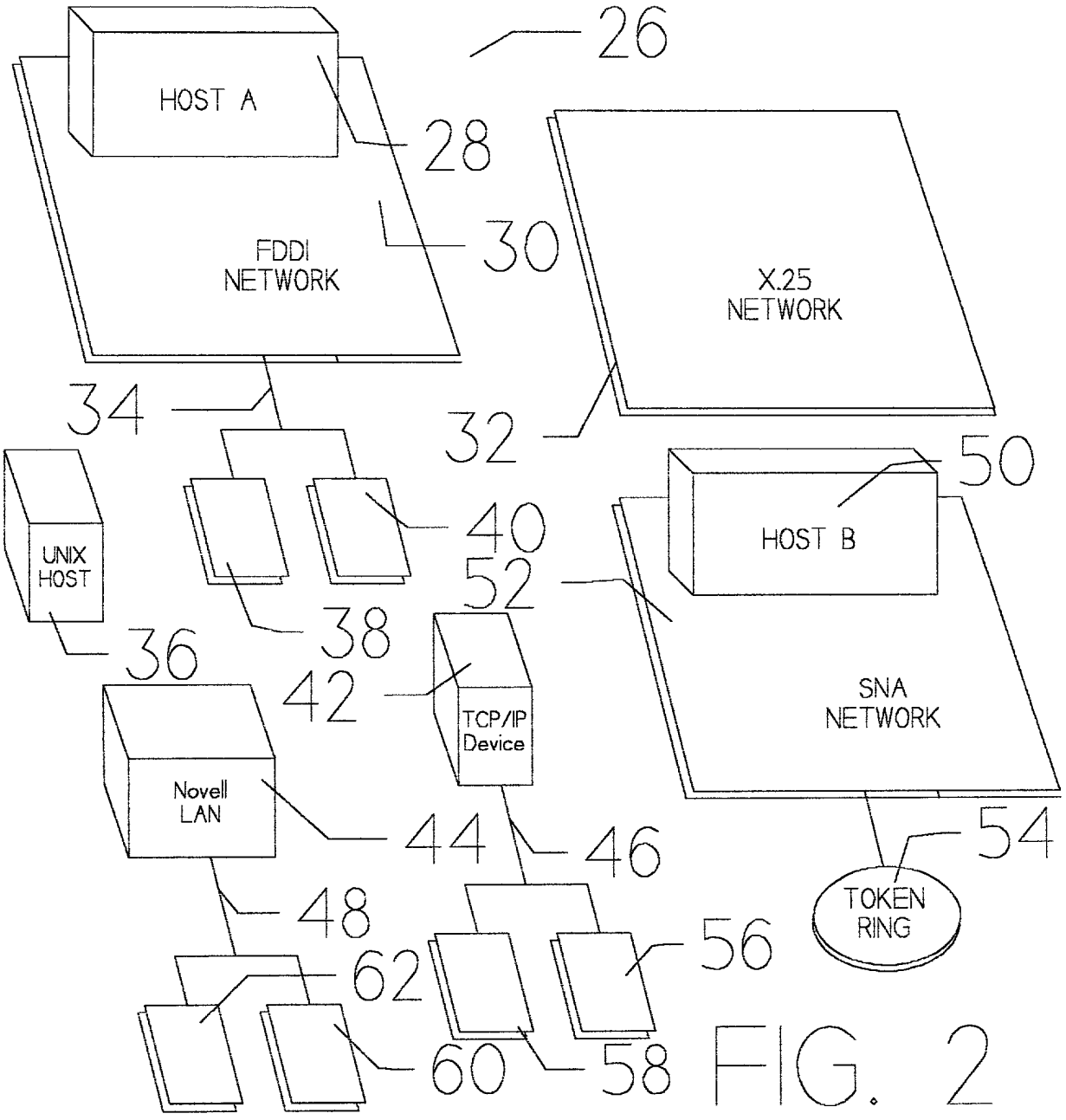
26
HOST A
28
FDDI
NETWORK
30
X.25
NETWORK
34
32
50
UNIX
HOST
40
HOST B
52
38
36
42
TCP/IP
Device
SNA
NETWORK
Novell
LAN
44
46
54
48
TOKEN
RING
56
62
60
58

FIG. 2

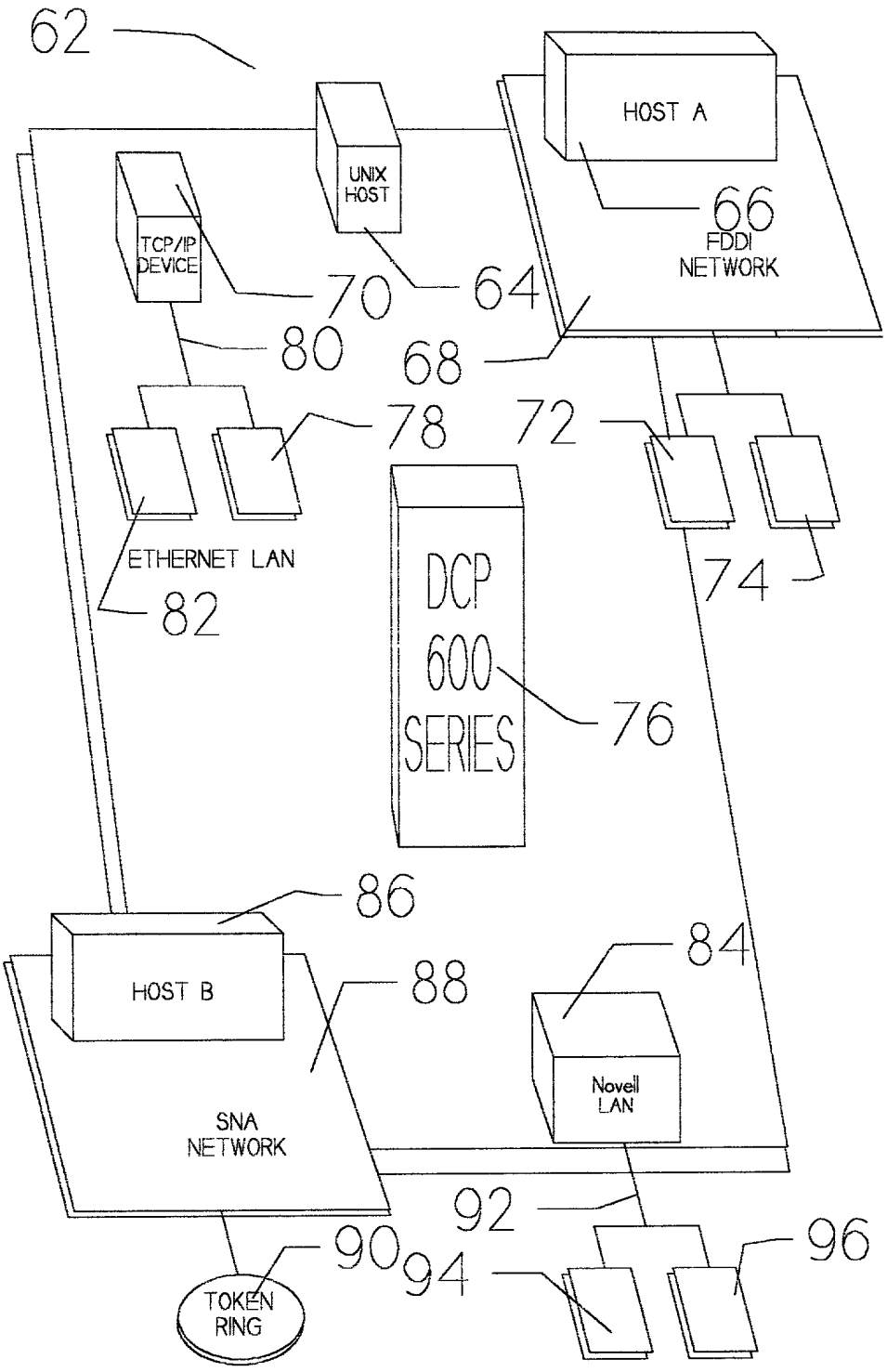
62
HOST A
UNIX HOST
66
FDDI NETWORK
TCP/IP DEVICE
70
64
80
68
78
72
ETHERNET LAN
82
DCP 600 SERIES
76
74
86
HOST B
88
84
Novell LAN
SNA NETWORK
92
90
94
96
TOKEN RING

FIG. 3

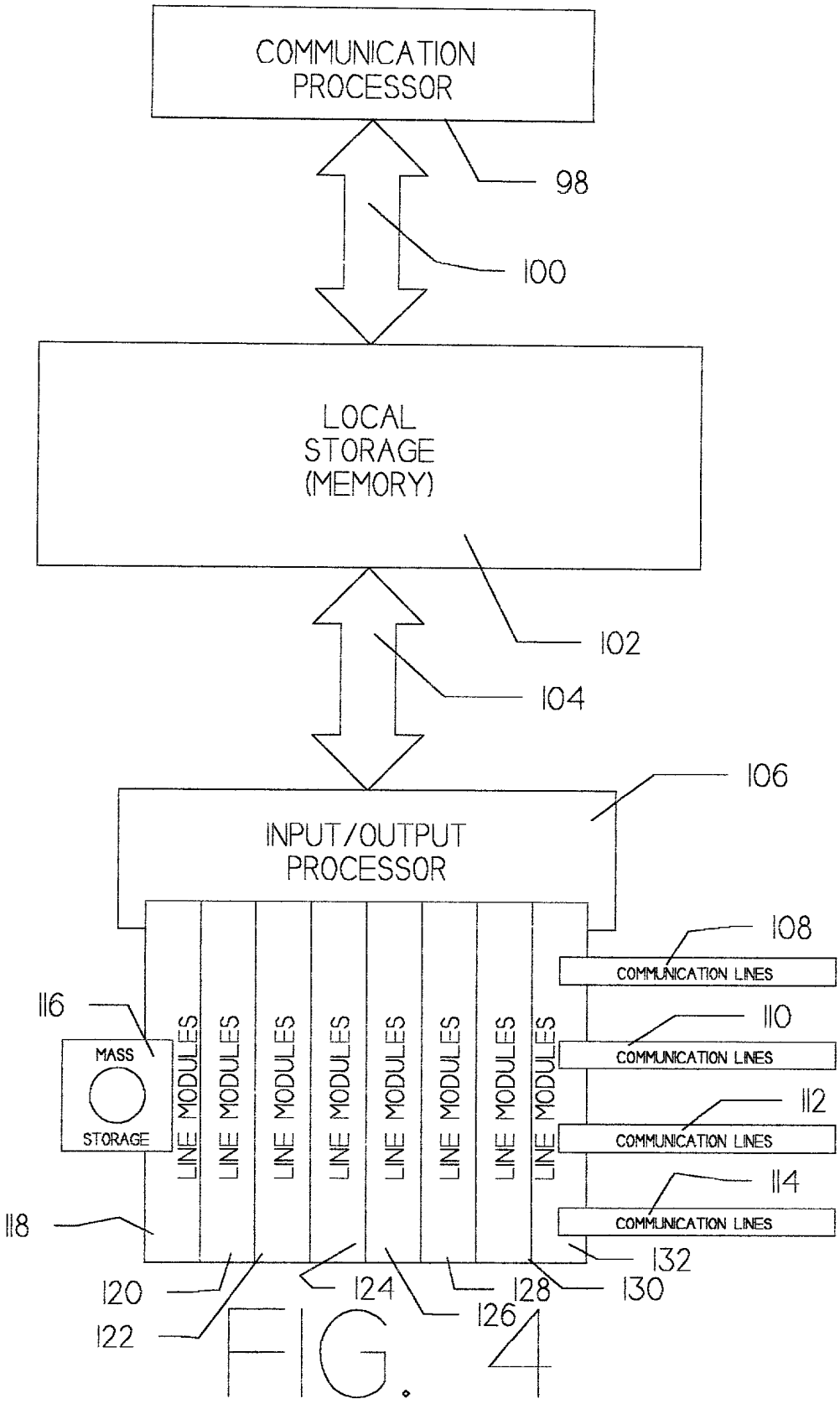
COMMUNICATION PROCESSOR
98
100
LOCAL STORAGE (MEMORY)
102
104
106
INPUT/OUTPUT PROCESSOR
108
COMMUNICATION LINES
110
COMMUNICATION LINES
112
COMMUNICATION LINES
114
COMMUNICATION LINES
116
MASS STORAGE
LINE MODULES
LINE MODULES
LINE MODULES
LINE MODULES
LINE MODULES
LINE MODULES
LINE MODULES
118
120
122
124
126
128
130
132

FIG. 4

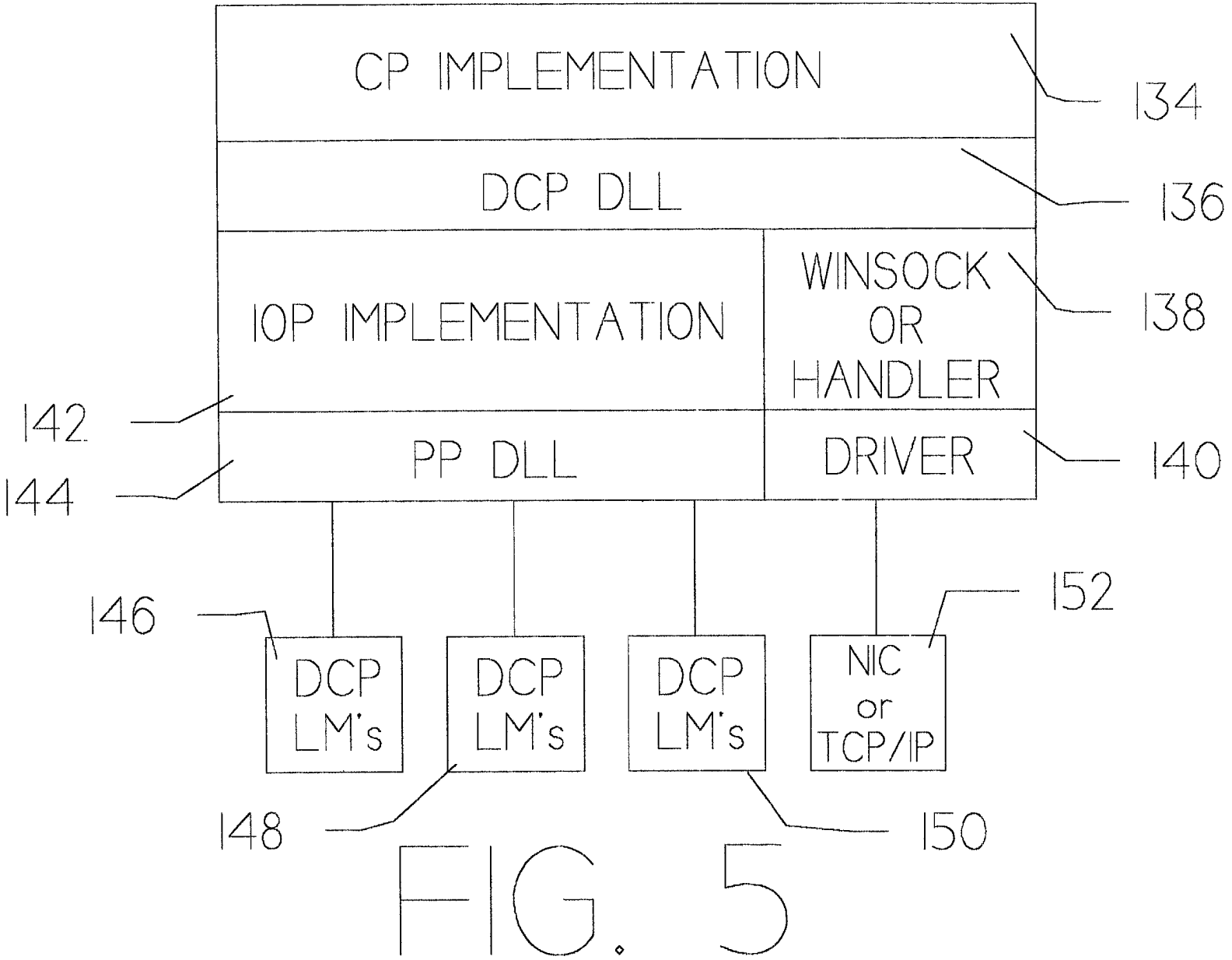

FIG. 5

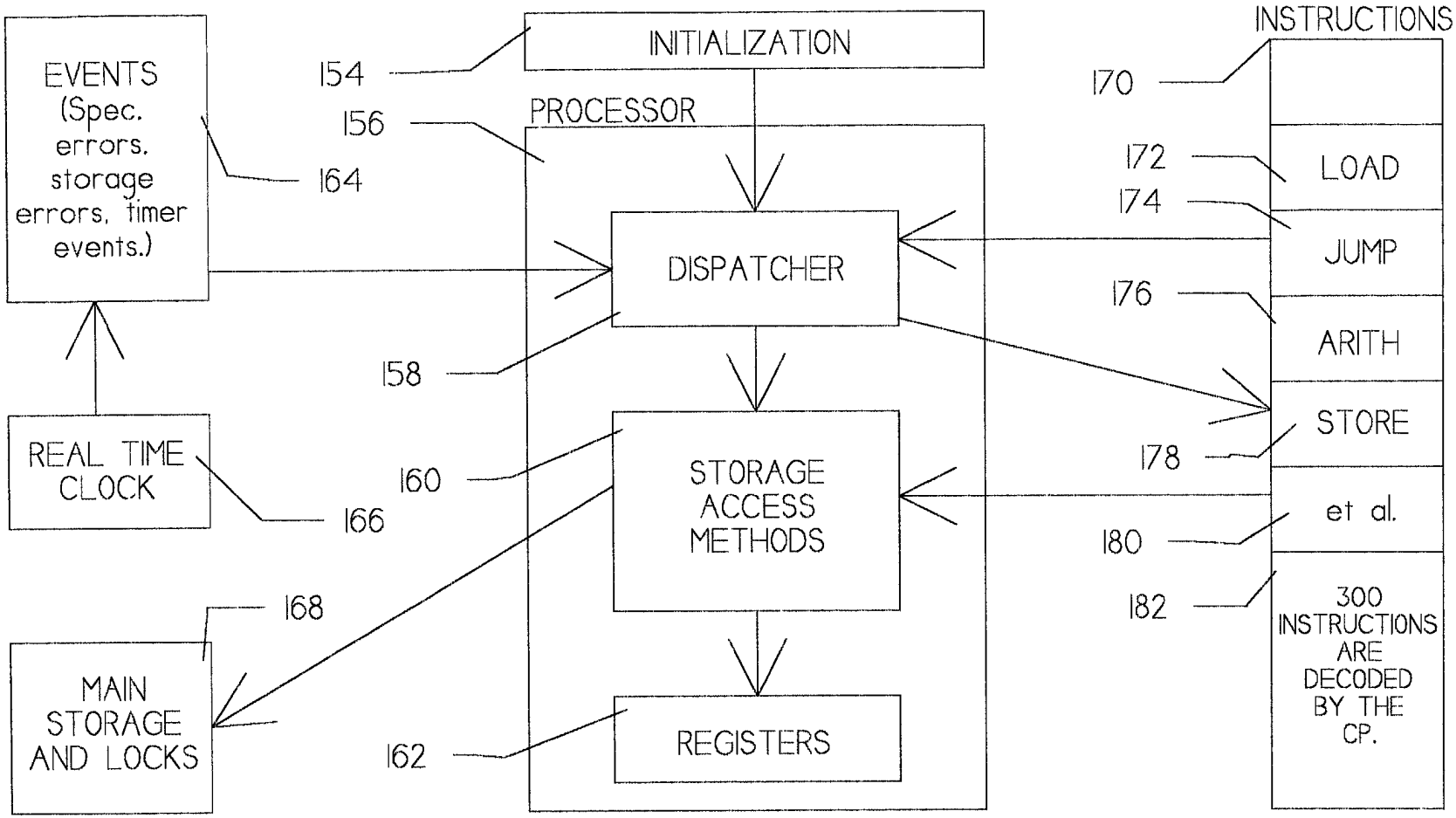

FIG. 6

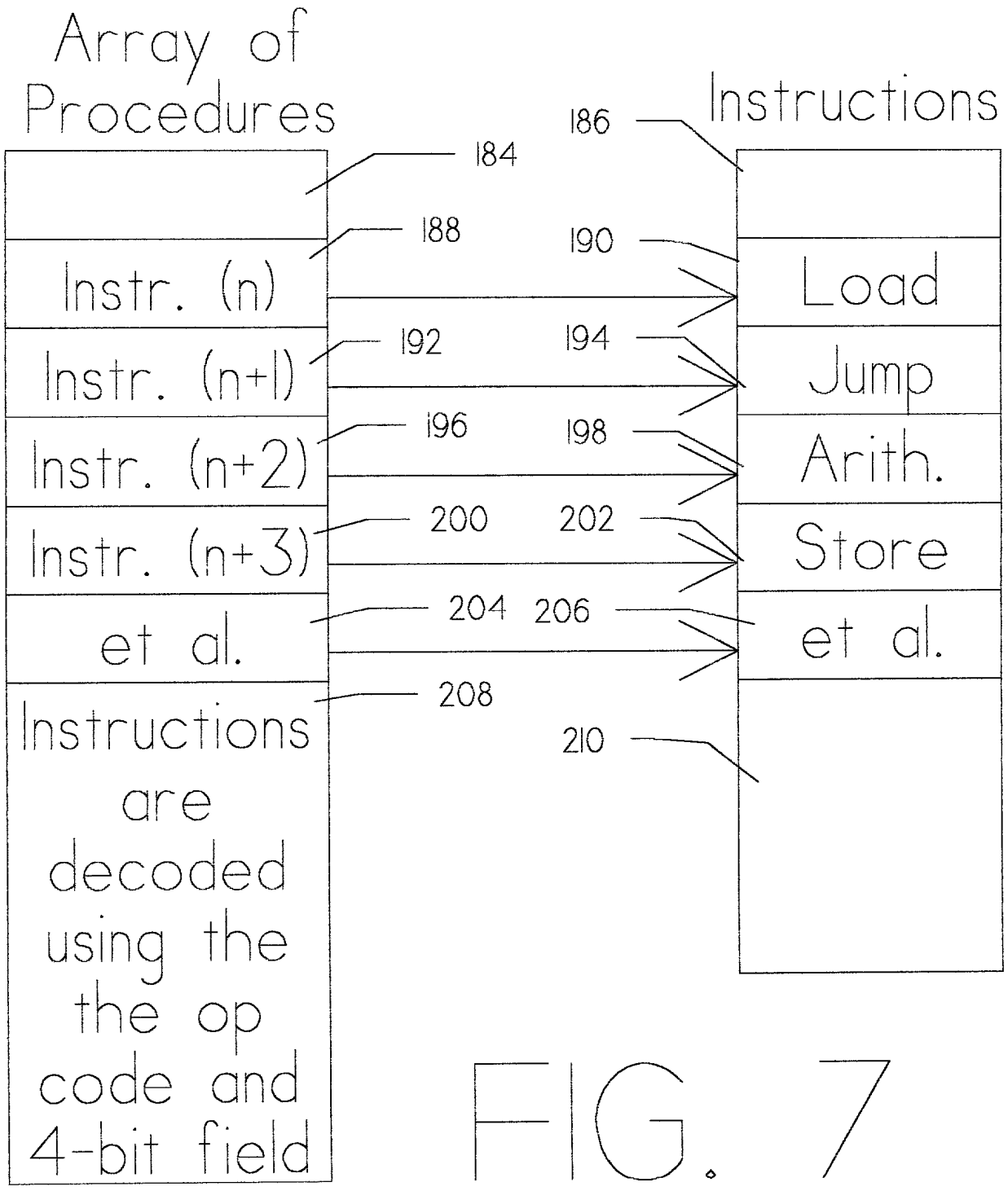
Array of Procedures
Instructions
184
186
188
190
Instr. (n)
Load
192
194
Instr. (n+1)
Jump
196
198
Instr. (n+2)
Arith.
200
202
Instr. (n+3)
Store
204
206
et al.
et al.
208
Instructions are decoded using the the op code and 4-bit field
210

FIG. 7

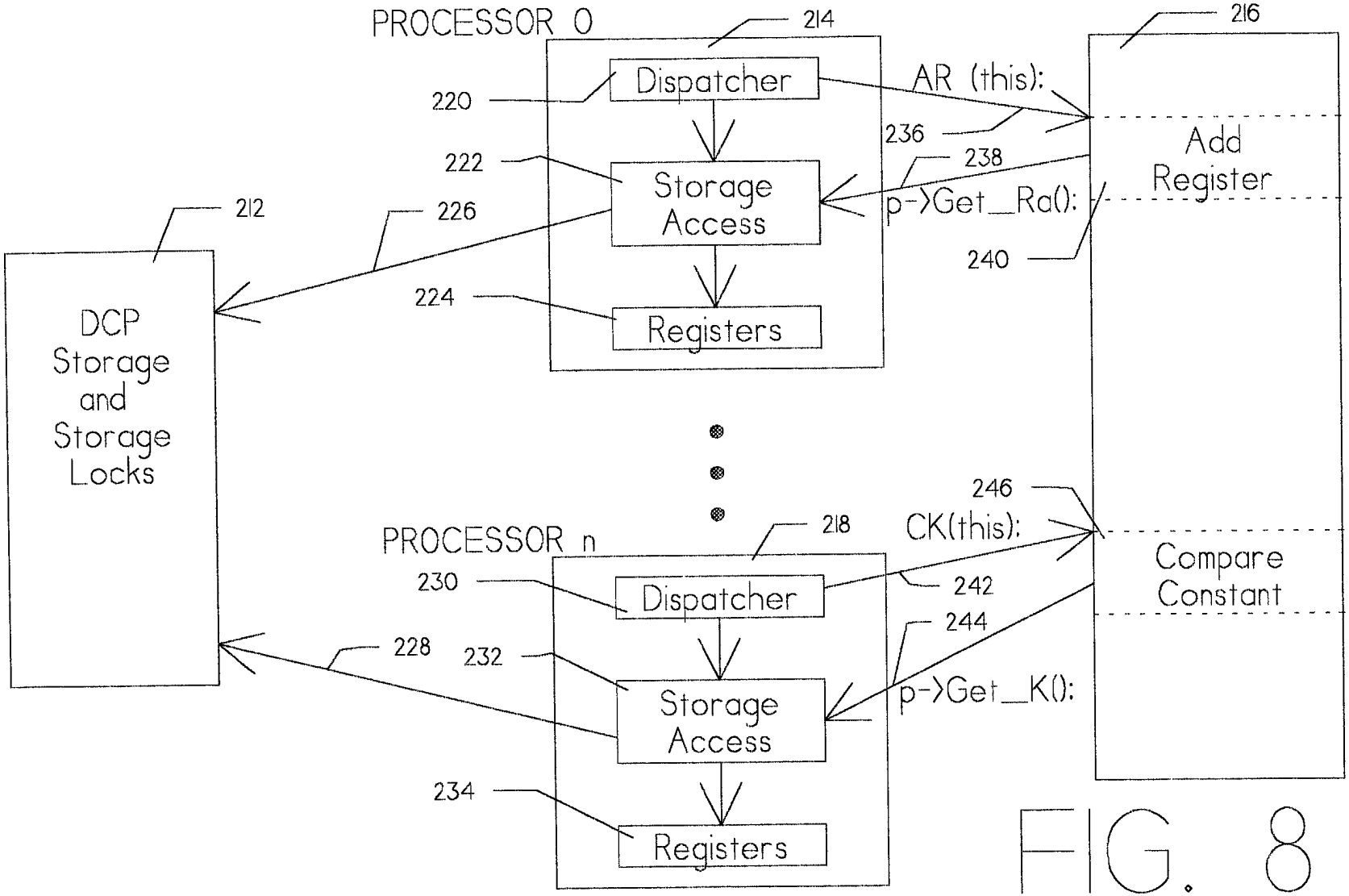

FIG. 8

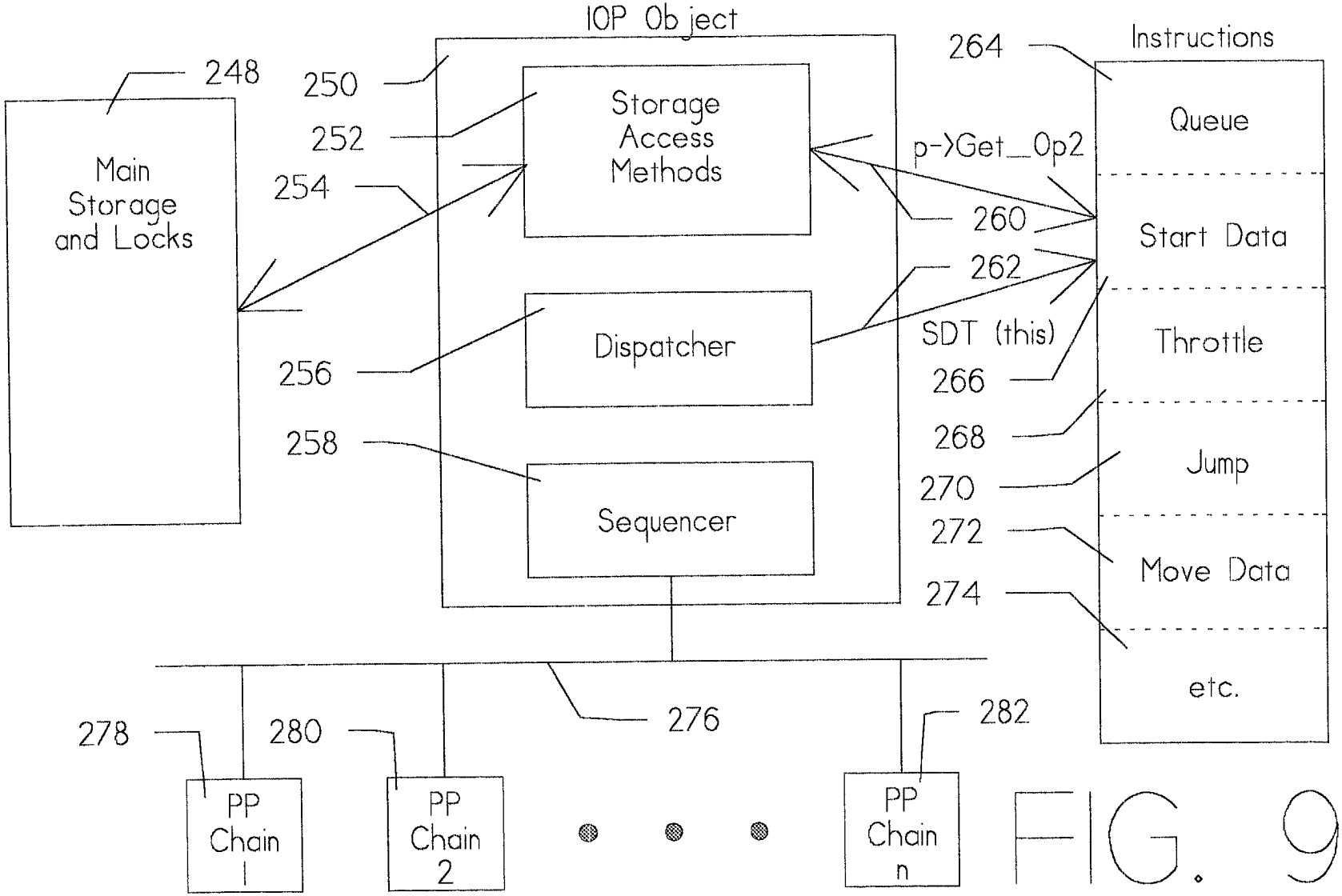

FIG. 9

COMBINED DECLARATION/POWER OF ATTORNEY FOR PATENT APPLICATION

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name.

I believe that I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled METHOD FOR EMULATING MULTI-PROCESSOR ENVIRONMENT, the specification of which (check one)

XX is attached hereto

__ was filed on _____
as U.S. Application
Serial No. _____

__ and was amended on (if
applicable) _____

I hereby state that I have reviewed and understand the contents of the above-identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to the examination of this application in accordance with Title 37, Code of Federal Regulations, §1.56(a).

I hereby claim foreign priority benefit(s) under Title 35, United States Code §119 of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application(s) for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

Prior Foreign Application(s)			Priority Claimed	
(Number)	(Country)	(Day/Month/Year Filed)	YES	NO
(Number)	(Country)	(Day/Month/Year Filed)	YES	NO
(Number)	(Country)	(Day/Month/Year Filed)	YES	NO

I hereby claim the benefit under Title 35, United States Code, §120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner

provided by the first paragraph of Title 35, United States Code, §112, I acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, §1.56(a) which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

(Serial No.)	(Filing Date)	(Status) (patented, pending, abandoned)
(Serial No.)	(Filing Date)	(Status-patented, pending, abandoned)

POWER OF ATTORNEY: As a named inventor, I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and transact all business in the Patent and Trademark Office connected therewith.

John L. Rooney, Reg. No. 28,898;
Lawrence M. Nawrocki, Reg. No. 29,333;
Wayne A. Sivertson, Reg. No. 25,645;
Richard C. Stempkovski, Jr., Reg. No. 45,130;
Jeffery L. Cameron, Reg. No. 43,527;
Donald A. Jacobson, Reg. No. 22,308; and
Charles A. Johnson, Reg. No. 20,852

Send correspondence to:

Charles A. Johnson
Unisys Corporation
Law Department
M.S. 4773
2470 Highcrest Road
Roseville, Minnesota 55113

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon, I further declare that I understand the content of this declaration.

Full name of sole or first inventor John J. Burns
Inventor's Signature _____ Date 2/28/2000
Residence 929 East Indian Springs Road
Bountiful, Utah 84010-3337 Citizenship U.S.A.
Post Office Address 929 East Indian Springs Road
Bountiful, Utah 84010-3337

1.56 Duty to disclose information material to patentability.

(a) A patent by its very nature is affected with a public interest. The public interest is best served, and the most effective patent examination occurs when, at the time an application is being examined, the Office is aware of and evaluates the teachings of all information material to patentability. Each individual associated with the filing and prosecution of a patent application has a duty of candor and good faith in dealing with the Office, which includes a duty to disclose to the Office all information known to that individual to be material to patentability as defined in this section. The duty to disclose information exists with respect to each pending claim until the claim is cancelled or withdrawn from consideration, or the application becomes abandoned. Information material to the patentability of a claim that is cancelled or withdrawn from consideration need not be submitted if the information is not material to the patentability of any claim remaining under consideration in the application. There is no duty to submit information which is not material to the patentability of any existing claim. The duty to disclose all information known to be material to patentability is deemed to be satisfied if all information known to be material to patentability of any claim issued in a patent was cited by the Office or submitted to the Office in the manner prescribed by §§1.97(b)-(d) and 1.98. However, no patent will be granted on an application in connection with which fraud on the Office was practiced or attempted or the duty of disclosure was violated through bad faith or intentional misconduct. The Office encourages applicants to carefully examine:

(1) prior art cited in search reports of a foreign patent office in a counterpart application, and

(2) the closest information over which individuals associated with the filing or prosecution of a patent application believe any pending claim patentably defines, to make sure that any material information contained therein is disclosed to the Office.

(b) Under this section, information is material to patentability when it is not cumulative to information already of record or being made of record in the application, and

(1) It establishes, by itself or in combination with other information, a prima facie case of unpatentability of a claim; or

(2) It refutes, or is inconsistent with, a position the applicant takes in:

(i) Opposing an argument of unpatentability relied on by the Office, or
(ii) Asserting an argument of patentability.

A prima facie case of unpatentability is established when the information compels a conclusion that a claim is unpatentable under the preponderance of evidence, burden-of-proof standard, giving each term in the claim its broadest reasonable construction consistent with the specification, and before any consideration is given to evidence which may be submitted in an attempt to establish a contrary conclusion of patentability.

(c) Individuals associated with the filing or prosecution of a patent application within the meaning of this section are:

(1) Each inventor named in the application:

(2) Each attorney or agent who prepares or prosecutes the application; and

(3) Every other person who is substantively involved in the preparation or prosecution of the application and who is associated with the inventor, with the assignee or with anyone to whom there is an obligation to assign the application.

(d) Individuals other than the attorney, agent or inventor may comply with this section by disclosing information to the attorney, agent, or inventor.